1F

(12) UK Patent Application (19) GB (11) 2 062 314 A

(21) Application No 8034967
(22) Date of filing 30 Oct 1980
(30) Priority data
(31) 89493
(32) 30 Oct 1979
(33) United States of America (US)
(43) Application published 20 May 1981
(51) INT CL³ G06F 13/00
(52) Domestic classification G4A FM
(56) Documents cited None
(58) Field of search G4A
(71) Applicant Honeywell Inc., Honeywell Plaza, Minneapolis, Minnesota 55408, United States of America
(72) Inventors Steven A. Rose, Edward H. Forrester
(74) Agent John Riddle & M. G. Harman individually, Honeywell Control Systems Ltd., Patent Department, Charles Square, Bracknell, Berkshire

(54) Protection Apparatus for Multiple Processor Systems

(57) In a data protection apparatus for a multiple CPU system having a common or multiported bulk memory, an interface structure 6A, 6B, etc., is associated with each of the CPU's 2A, 2B etc. The interface structure cooperates with a firmware engine 32 which is, in turn, a part of the interface control means 28 which controls the transfer of data between the common bulk memory apparatus and each of the CPU's in the system. Signals generated by the individual CPU's indicative of emergencies (e.g. proven failure or buffer full) are fed to the interface structures. The interface structures then translate those signals into an attention flag signal (from 36A, etc) on bus 52 and signals (from status latch register 38A, etc on bus 54) identifying the source or nature of the emergency. The firmware engine 32 repeatedly interrogates bus 52 to see if a attention signal is present. If it is, the engine 32 then interrogates bus 54 to identify the nature of the emergency, determines which CPU is affected, and effects the necessary measures to protect the data relative to the affected CPU.

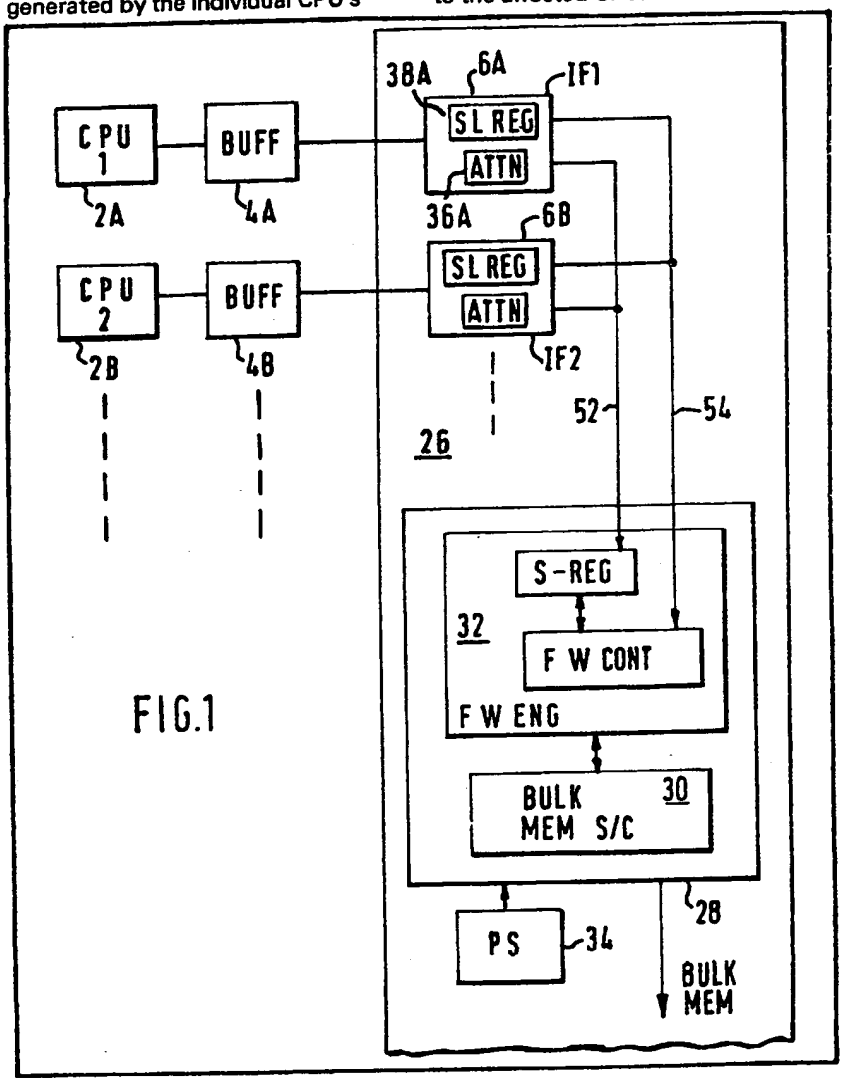

FIG.1

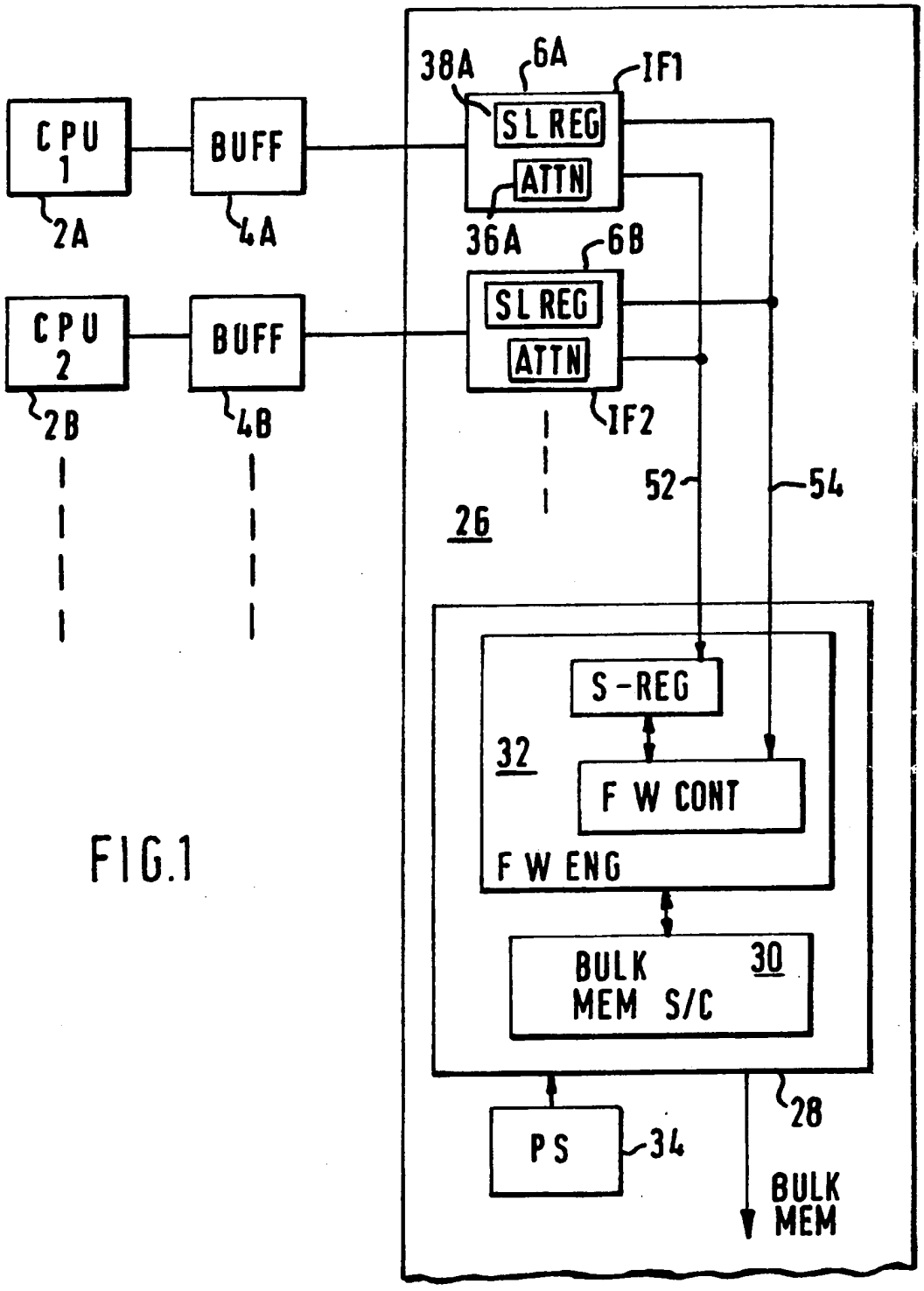
CPU 1
BUFF
2A
4A
CPU 2
BUFF
2B
4B
38A
6A
IF1
SL REG
ATTN
36A
6B
SL REG
ATTN
IF2
52
54
26
S-REG
32
F W CONT
F W ENG
BULK MEM S/C
30
PS
34
28
BULK MEM

FIG.1

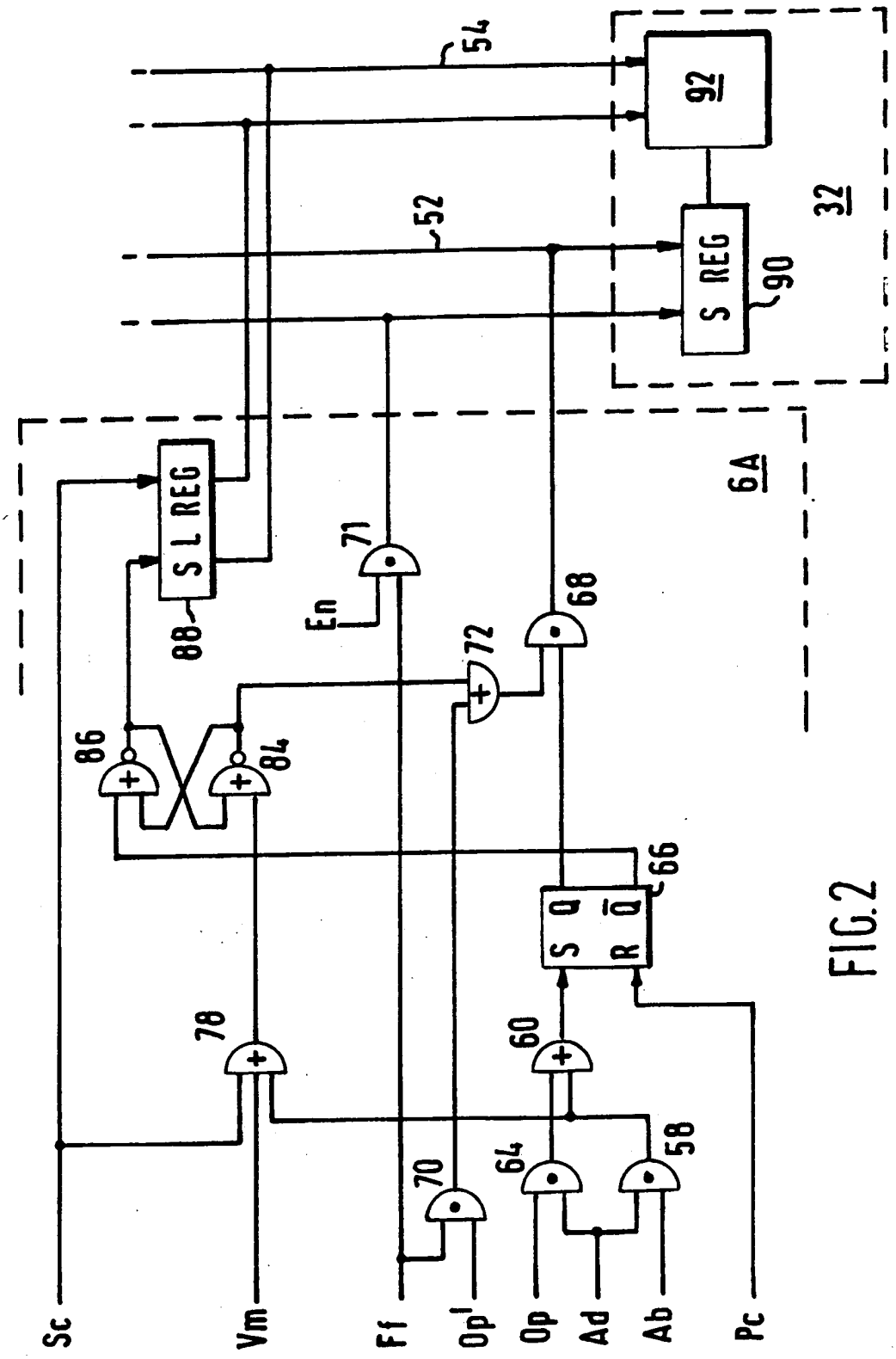
54
92
52
S REG
90
32
6A
S L REG
88
71
En
68
72
86
84
66
S Q
R Q̄
78
60
70
64
58
Sc
Vm
Ff
Op'
Op
Ad
Ab
Pc

FIG.2

SPECIFICATION

Protection Apparatus for Multiple Processor Systems

The present invention relates to computer systems, and more particularly to a protective interrelation means in multiple processor systems.

In some computer based control systems, a plurality of central processor units (CPU's) may work in conjunction with a common bulk memory means. In such an arrangement, there will, of necessity, be interface control means for controlling the access of the several CPU's to the common memory. It is a normal function of such interface control means to control the transfer of data between the bulk memory and the several CPU's under a normal priority routine. If, however, there occurs an emergency situation in one or more of the CPU's, valuable data may be lost or incorrect data transferred in the absence of protective measures to avoid said eventuality. For example, if the power supply should fail in one of the CPU's, there is a time frame of 1 ms in which all transactions relating to that CPU must be accomplished in order to preserve accurate data relative to that CPU.

It is, accordingly, an object of the present invention to provide protection means responsive to emergency conditions at any of the CPU's of a multiple CPU system.

In accordance the present invention provides a computer system comprising a plurality of processors coupled by respective interface units to a bulk memory unit and a multiported memory control unit which controls transactions between the bulk memory unit and the processors, each processor producing a plurality of condition signals indicative of conditions requiring attention, each interface unit being responsive to the condition signals from the associated processor to produce an attention signal indicative of the presence of any received condition signals and a status signal indicative of the nature of any received condition signal, the control unit having first and second buses to which the attention signals and status signals, respectively, of all interface units are coupled in common, and repeatedly interrogating the first bus to determine whether any interface unit is producing an attention signal and, if so, interrogating the second bus to identify the condition requiring attention and then controlling the interaction of the affected processor and the bulk memory unit so as to protect the data relating to the affected processor.

A system embodying the present invention will now be described, by way of example, with reference to the drawings, in which:

Figure 1 is a block diagram of a multiple CPU system, and

Figure 2 is a logic diagram of parts of the system.

Figure 1 shows a computer system wherein a plurality of CPU's are connected to share the services of a common bulk memory structure. A first CPU 2A is connected through a port buffer 4A to a first interface unit 6A. Further CPU's such as 2B are connected through corresponding buffers and interface units as shown. Each of the buffers is closely associated with the corresponding CPU, and each of the interface units is mounted in a bulk memory controller cabinet 26.

The cabinet 26 also contains a bulk memory controller 28 which includes a bulk memory subcontroller 30 and a firmware engine 32. A common power supply unit 34 supplies power to the bulk memory controller system and the interface units. The interface units, the firmware engine 32, and the bulk memory subcontroller 30 are interconnected as by a cascade ribbon interconnect cable. The buffers each include a plurality of line drivers and receivers and provide a measure of load buffering between the corresponding CPU's and the associated interface units. The firmware engine 32 in the bulk memory controller 28 comprises a microprocessor together with its predetermined operating instructions stored in its internal memory. The bulk memory subcontroller 30 controls the actual interchange of information between the several CPU's and a bulk memory apparatus (not shown). The operation of the bulk memory subcontroller 30 is also directed by the firmware engine 32.

In each of the interface units 6A, 6B, etc., there is an attention logic arrangement which responds to certain output signals from the associated CPU-s indicating predetermined conditions at the associated CPU. Thus interface unit 6A includes an attention unit 36A which provides an output signal whenever the associated CPU is in one of the predetermined conditions just mentioned. A status register 38A in the first interface unit 6A is arranged to include data which would be indicative of the particular one of the predetermined conditions. The other interface units have similar attention units and status registers. The significant positions in the registers of all the interface units are connected in common to one input of the firmware engine 32.

In operation of the system the firmware engine 32, under the control of its internal instructions, controls the transfer of data between the several CPU's and the multiported bulk memory. As a part of that operation, the common line 52 is examined on a periodic basis to determine if there is an attention signal present on the signal significant bit position of any of the registers 36A, 36B, etc. Depending on the circumstances, the frequency of that periodic testing the line 52 may occur at a rate which approaches the range of once every 500 ns. If there is no attention signal present at any of the interface units 6A, 6B, etc., the firmware engine returns to its ordinary operation of controlling the transfer of data to and from the several CPU's in accordance with their established priority. If, on the other hand, an attention signal is present at one of the interface units the firmware engine then scans the input

terminals thereto represented by the cable 54 to determine, first, which of the interface units is carrying the attention flag and, second, what type of emergency caused the flag to be up. When the affected CPU has been identified, the firmware engine immediately completes any transaction scheduled relative to that CPU. The firmware engine then determines if there are other attention flags active and, if so, identifies and completes transactions relating to those CPU's. When all attention flags have been acknowledged and handled, the firmware engine returns to its basic task of servicing requests for bulk transfer.

Figure 2 shows the detailed logic of the attention logic arrangement of the interface unit 6A; all the interface units are substantially identical. The interface unit 6A, it will be recalled, is interconnected between the CPU 2 by way of the buffer 4 and the bulk memory subcontroller 30. Thus the input signals to the attention logic system of the interface unit 6A are received by way of a multiconductor bus from the CPU 2A. Corresponding input signals would be applied to the input of the other interface units from their associated CPU's.

An operate signal Op may be any one of a group of signals indicating a particular operation for the transfer of data between the CPU and the bulk memory. By design that signal is mutually exclusive with respect to an abort signal Ab. An address signal Ad is indicative that the CPU concerned has been addressed, and acts as an enable signal for the signals Op and Ab, allowing them to pass through respective AND gates 64 and 58 and an OR gate 60 to set a busy flip-flop 66. Thus the coincidence of the address signal Ad and an operate signal Op is effective to set the flip-flop 66 to indicate that the interface unit is busy. The Q output of the flip-flop 66 enables a gate 68.

In accomplishing a transfer of data between the CPU and the bulk memory, there is provided in each of the interface units a first-in first-out (FIFO) memory stack (not shown) in which data and/or instructions are stored on a temporary basis during the course of the transfer. A signal Op' indicates when the FIFO is being addressed. If the FIFO is full and can accommodate no new entries, a signal Ff is generated and fed to a line driver gate 71. The driver 71 is enabled by an enable signal En whenever the particular unit is addressed by the firmware engine. The signal Ff is also combined with the signal Op3 in a gate 70 which feeds a gate 68 via an OR gate 72. Gate 68 is also fed with the Q (set) output of the busy flip-flop 66, and its output corresponds to the output of the attention unit 36 on the line 52, Figure 1, which is common to all the interface units and is connected to the firmware engine 32. If, while the gate 68 is enabled, the Op' signal, which is used to address the FIFO, coincides with the signal Ff indicating that the FIFO is full, gate 70 passes the combined signal via gate 72 to the attention driver 68. This signifies to the firmware engine that the particular channel represented by the indicated interface unit and its associated CPU needs attention from the firmware engine and the bulk memory.

The attention signal output by the driver 68 is loaded into bit position No 7 of an S-register 90. Similarly, the output of the line driver 71 is loaded into bit position No 6 of the Sregister 90. A firmware control unit 92 periodically interrogates the S-register to determine if there is an attention signal present at the output of any of the interface units. If there is, the firmware engine then sequentially addresses the interface units to determine which of them requires attention and then to determine the nature of the attention required. In the operation discussed thus far, when the interface unit 6 is addressed, the line driver 71 is enabled and the signal indicating the not empty condition of the FIFO on that board is lodged into bit position No. 6. This signal is recognized by the firmware engine and a subroutine is initiated to read data out of the indicated FIFO into certain other storage registers to partially empty the FIFO, thereby to enable the insertion of the new data therein. If that had been the only requirement for attention at the addressed interface unit, the firmware engine would issue a port clear signal Pc which would then reset the busy flip-flop 66 and thereby remove the attention signal from the output of the driver 68. The firmware engine would then return to the routine in which it had been engaged before the diversion to the attention routine.

The CPU may also transmit to the interface unit 6 the abort signal Ab which, when gated by the address signal Ad, is transmitted through the gates 58 and 60 to set the busy flip-flop 66 thereby enabling the driver 68. Gate 58 also feeds, via an OR gate 78, a latch formed by two gates 86, 84. The SET output of the gate 84 is applied via the OR gate 72 to the driver 68. This combination also will set the attention signal into the bit 7 position of the S-register 90.

Gate 86 feeds a status latch register 88 and sets the bit 7 position. When the firmware engine 32 has detected that an attention signal is present in the S-register 90, it again polls the several interface units sequentially and "enables" an output from the status latch 88 to identify that the particular interface unit is the one calling for attention and identifies the nature of the call for attention. When the firmware engine has thus recognized the identifying signal in the No. 7 bit position of the status latch 88, a PORT CLEAR routine is instituted which causes any required data transfer between the indicated CPU and the bulk memory to be immediately completed. The firmware engine then issues a port clear signal Pc which is applied to reset the BUSY flip-flop 66, thereby resetting the latch 86, 84 and disabling the driver 68.

Another signal that may be generated by this CPU is a voltage monitor alarm signal Vm from a voltage monitor indicating that there has been a failure in the power supply of that CPU. That signal is supplied via the OR gate 78 to set the

latch 84, 86. The operate signal Op will, as described above, enable the output driver 68. The voltage monitor signal Vm, having set the latch 84, 86, will cause the driver 68 to produce an attention signal via OR gate 72. The latch 84, 86 also registers the signal in the bit 7 position of the status latch register 88. As before, the firmware engine terminates any ongoing data transfers and then intiates the PORT CLEAR routine, which clears all outstanding requests from that port and sets the port not busy. Software in the CPU recognizes when this has occurred and proceeds to sequence the power-removal from the computer. This transaction is essential before the actual loss of power at the CPU occurs. The interval between the occurrence of the signal Vm and the total loss of power from the CPU, as hereinbefore noted, is about 1 ms. The actual transfer of the data between the CPU and the bulk memory takes a small portion of that time. Accordingly, the frequent scanning by the firmware engine of the bit 7 position in the S-register to detect the presence or absence of such an attention signal provides ample time for the completion of the necessary transfers before the loss of power causes potentially erroneous alteraction of the stored data.

Still another signal that may be generated by the CPU is a system clear signal Sc which is fed to the OR gate 78. As before, the Sc signal sets the latch 84, 86, thereby establishing the attention signal at the output of the NAND gate 68 and setting the bit 7 position of the S-register 90. Also the setting of the latch 84, 86 sets the bit 7 position of the status latch 88. As before, the firmware engine checks the S-register for the attention signal then checks to determine the source of the attention signal, followed by the PORT CLEAR routine. The Sc signal also sets bit 5 of the status latch 88. That signal is also read by the firmware engine and institutes a different routine which is not relevant to present purposes.

The relevant parts of the subroutine or instruction sequence executed by the firmware engine will now be described. This can be analyzed into three major portions, I to III.

Portion I

As an initial step the firmware engine will read the S-register to determine if bit 7, the attention signal, is set. If no attention signal is set the firmware engine will return to the routine upon which it was operating at the time that it paused to read the S-register. If the bit 7 position is set, then the firmware engine will read the status latch, first to determine if the 7 or 5 bit position is set. If neither is set, a subroutine ATTENTION 2, portion II, is initiated.

Portion II

The first interface unit or port is examined to determine if the Op' signal is true. If, in the selected port, the Op' signal is false then a check will be made of the second and succeeding ports. If the Op' signal is set, the attention flag is indicative that the FIFO contains data requiring immediate (time-critical) service. Therefore, the firmware engine will cause the FIFO to be read and the data to be processed immediately. If after the first reading of the FIFO, the FIFO still contains data, then the process will be repeated until the FIFO is empty. When it is determined that the FIFO is not empty, then the firmware engine will issue a PORT CLEAR signal and return to its ongoing routine.

Portion III

If, on reading the status latch at the end of portion I, it is found that either the bit 7 position or the bit 5 position is set, then the status latch is interrogated to determine which of the bit positions is set. If the bit 5 position is set a subroutine which is not relevant for present purposes will be instituted. If the bit 7 position of the status latch is set, the firmware engine will initiate a PORT CLEAR routine which includes the termination of any data transfer in progress on the present port. Than a PORT CLEAR signal will be issued (as at the end of portion II) and the firmware engine will return to its basic task of servicing requests for bulk transfers.

Claims

1. A computer system comprising a plurality of processors coupled by respective interface units to a bulk memory unit and a multiported memory control unit which controls transactions between the bulk memory unit and the processors, each processor producing a plurality of condition signals indicative of conditions requiring attention, each interface unit being responsive to the condition signals from the associated processor to produce an attention signal indicative of the presence of any received condition signals and a status signal indicative of the nature of any received condition signal, the control unit having first and second buses to which the attention signals and status signals, respectively, of all interface units are coupled in common, and repeatedly interrogating the first bus to determine whether any interface unit is producing an attention signal and, if so, interrogating the second bus to identify the condition requiring the attention and the controlling the interaction of the affected processor and the bulk memory unit so as to protect the data relating to the affected processor.

2. A system according to Claim 1, wherein each processor has an associated memory check unit and produces a condition signal on its stack becoming full, and the control unit thereupon causes reading from the stack until the stack is empty.

3. A system according to either previous claim, wherein each processor produces a condition signal on the occurrence of a power supply failure, and the control unit thereupon causes the termination of any transfer of data to or from the affected processor before the processor becomes inoperative.

4. A system according to any previous claim, wherein the control unit, on detecting an attention signal, polls the interface units in sequence and monitors the second bus to identify which unit is producing the attention signal.

5. A computer system substantially as herein described and illustrated.

Printed for Her Majesty's Stationery Office by the Courier Press, Leamington Spa, 1981. Published by the Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.